An Online Framework for Diagnosis of Multiple Defects in Scan Chains

Sarmad Tanwir*, Michael S. Hsiao* and Loganathan Lingappan†
*Virginia Tech, Blacksburg, VA, USA email: {sarmadt, mhsiao}@vt.edu
†Intel Corporation, Folsom, CA, USA email: loganathan.lingappan@intel.com

***Abstract*—We propose a novel and effective online method for performing diagnosis of scan chains with the physical defective circuits in the loop. We first apply flush tests to determine the faulty chains and their corresponding fault types. Then, we generate new patterns using an evolutionary algorithm and quickly analyze the responses to perform diagnosis. We are able to achieve an average of 70% and 37% improvement in the diagnosis quality for the segmented and non-segmented scan chains respectively, as compared to a state-of-the-art offline industry tool, when 0 to 7 faults were randomly inserted in each scan chain. Our method does require additional tester time, which may be preferred to the computational, setup and overhead costs of the offline diagnosis, especially during the yield learning process.**

Keywords*—*Scan chain, Diagnosis, Online, Multiple Defects

I. Introduction

Scan based testing is the de facto industry standard for post silicon testing and diagnosis of large sequential circuits. In fact, scan chains routinely occupy roughly 30% of the area of modern digital chips [1]. However, none of the scan based testing can proceed until the scan chains are healthy. Especially during the low-yield regime after the introduction of a new technology, the scan chains are frequently defective. Therefore, identification of failure locations on the scan chains is the predominant step in the testing and diagnosis process flow. Also, since the scan chains are usually distributed evenly all across the chip, chain diagnosis can be a significant contributor to the yield learning.

The authors of [2] have summarized about two decades of work on scan chain diagnosis. They classify the techniques developed so far into three categories: tester based, hardware based and software based. In the tester based techniques, special equipment such as an electron beam probe [3] or picosecond imaging [4] may be integrated with the tester to perform a binary search for the failing scan cells. The diagnostic resolution of these methods is good; however, time spent in the binary search with the specialized equipment can be prohibitive.

The hardware based techniques employ partner shift registers [5], XOR gates inserted between cells [6] or additional scan cell set/reset circuitry [7], [8] to enhance the diagnosis quality. These techniques have not been widely adopted by the industry because they need additional hardware in or around *each* scan cell. Therefore, there is a high hardware overhead that is also prone to failure.

The software based techniques are more widely applied because no design modification is needed. These can be further classified into simulation based, probability based and dictionary based. The classic simulation based method proposed by Guo and Venkataraman [9], [10] makes use of the simulation invariant bits in scan chains to identify the upper and lower bounds of scan-chain failures. Their technique was further enhanced by Kao et al. [11] by 'jump simulation' wherein additional patterns are simulated to improve these bounds. Huang et al. [12], [13] proposed other methods that rely on backtracing the circuit graph from the observed mismatches on good chains to improve the bounds on the scan chain failures. More recently, Guo et al. [14] have proposed a dictionary-based method that uses differential signatures of scan cell failures to diagnose stuck-at, timing and multiple faults in a single scan chain.

In this paper, we present a hardware-in-the-loop method that generates and applies scan patterns to the circuit under diagnosis (CUD) to rapidly improve the failure lower bounds on scan chains with one or more stuck-at faults. It is a tester based method but, unlike [3], [4], without the hassle and cost of integrating the probe or imaging equipment. Our method adapts to each individual defective chip and generates the needed tests accordingly. We test our method on larger ISCAS'89 circuit models with multiple stuck-at faults on multiple scan chains and compare the achieved diagnosis quality with that of a state-of-the-art commercial tool.

In Section II, we elaborate on the limitations of the existing simulation based scan chain diagnosis methods and the motivation behind our work. In Section III, we describe our proposed scheme and its key advantages. Section IV lays out the detail design of our proposed online scan chain ATPG and diagnosis. In Section V and VI, we explain our experimental setup, evaluation method and the results of our experiments. Finally, we conclude the paper in Section VII.

II. Motivation

The automatic test equipment (ATE) for testing chips has advanced a lot since the pioneering Teradyne J259 and IBM 360 that were introduced around 1966. Since then, we have seen many improvements and introduction of new technologies such as memory testing (1973), scan testing (1977), parallel testing (1990's), GHz testing (1995) and test compression (early 2000's). Resultantly, today's ATEs are a lot faster, massively parallel and equipped with much smaller test heads. Our method for online diagnosis is another proposed technology improvement in this continuum of ATE evolution.

supported in part by NSF grant 1422054 and an Intel grant.

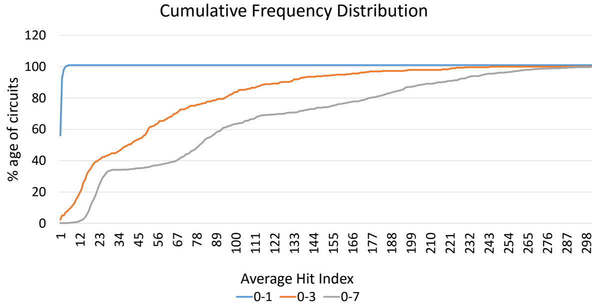

Fig. 1. Deterioration in Avg Hit Index with more Faults per Chain

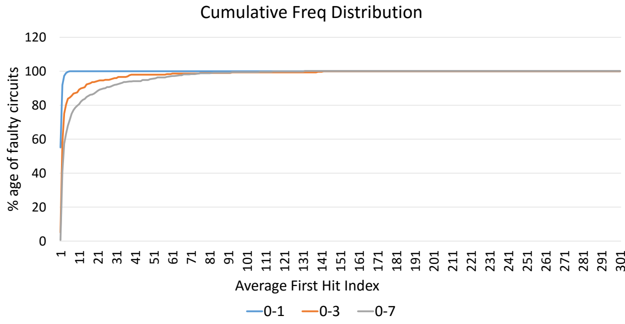

Fig. 2. Deterioration in Avg First Hit Index with more Faults per Chain

A. More faults per chain deteriorate diagnosis quality

Figures 1 and 2 show three graphs each that represent the diagnosis quality achieved using a state-of-the-art industrial tool for three populations of defective chips that have a maximum defect count of 1, 3 and 7 per scan chain. The diagnosis quality in these graphs and subsequent discussion is represented by two quantities.

1) Average Hit Index: The diagnosis tool reports a ranked list of suspects for each scan chain. The index of an actual fault location f in this list is called the *hit index* or $hi(f)$. We define the *average* hit index of a simulated faulty circuit instance as follows:-

$$hi_{avg} = \frac{\sum\limits_{\forall f \in F_{ins}} hi(f)}{|F_{ins}|}$$

where, F_{ins} is the set of all faults inserted in the circuit.

2) Average First Hit Index: The minimum of the hit indices of all faults inserted in a particular scan chain c is called the *first* hit index or $fhi(c)$. The average first hit index of a simulated faulty circuit instance is defined as:-

$$fhi_{avg} = \frac{\sum\limits_{\forall c \in C_f} fhi(c)}{|C_f|}$$

where, C_f is the set of all faulty chains in the circuit.

As explained in [15], these quantities are legitimate alternative representations of the *diagnostic resolution* when the reported suspects are *ranked*. The graphs in Figures 1 and 2 plot the cumulative frequency distribution of the average hit indices and the average first hit indices respectively. A point (x, y) on these graphs means that $y\%$ faulty chips were diagnosed with an average hi or average fhi of less than or equal to x for the stated number of inserted faults per chain. The higher the graph, the better is the diagnosis quality. We observe that while both the average hi and the average fhi deteriorate when more faults are inserted per chain, the deterioration in the average hi is more significant.

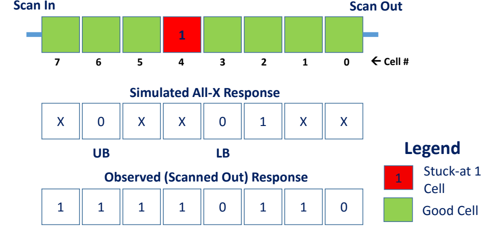

Fig. 3. Determining Bounds from Scan Chain Invariants

B. Limitations of Existing Methods

Methods proposed in [9]–[11] first simulate an all X scan pattern and find if any of the scan cells capture non-X responses. If the circuit is faulty free, the values of these scan cells will remain the same irrespective of the applied pattern. These *invariant* values help determine the bounds on the scan chain failure. For example, Figure 3 shows the response to the all X simulation and the actual response captured on the tester. As the value of cell 3 is observed 0 upon scan out, we can say that the stuck-at 1 fault in the scan chain can only exist upstream of cell 3. Also, the fault must either be at cell 6 or downstream of it to be able to change its value to 1 upon scan out. Hence, cell 6 is an upper bound. This upper bound does *not* hold under multiple fault assumption as there may be additional fault(s) upstream of cell 6.

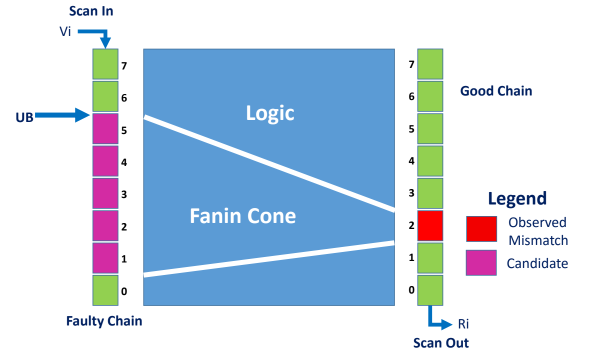

Fig. 4. Determining Bounds by Backtracing from Observation Points

In [12], the fundamental idea is to backtrace from an observed mismatch at a known good observation point. Figure 4 illustrates the idea. The mismatch observed at cell 2 of the good chain is traced back all the way to the input (faulty) scan chain. We conclude that this mismatch could not have propagated from scan cells 6 and 7 as they lie outside of the fanin cone of the mismatch. Therefore, cell 5 in the faulty chain is designated as the upper bound. However, under multiple fault assumption, there exists a possibility of scan cell 6 and 7 being defective such that the faults could not be propagated to the observation point(s) by the applied pattern.

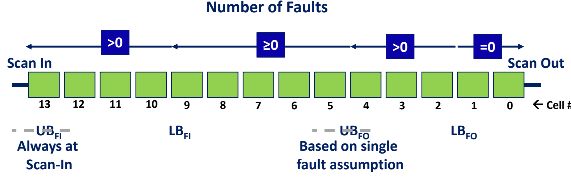

Fig. 5. Characteristics of Dominant Fault Pair Diagnosis

The multiple fault diagnosis technique in [13] computes two intervals for the dominant fault pair. They define the dominant fault pair as the two faults F_i and F_o that are closest to the scan chain input and output respectively. Each interval is defined by its corresponding upper and lower bound. The key characteristics of their diagnosis are as follows.

a) The lower bound of F_o is definitive, which means that it can be stated with certainty that no fault exists downstream of the lower bound. b) The upper bound of F_o means that there exists at least one fault downstream of it (until the lower bound of F_o). c) The lower bound of F_i means that there exists at least one fault upstream of it. d) The upper bound of F_i is always the cell closest to the scan chain input pin.

Figure 5 depicts the overall situation that emerges from this type of diagnosis. One may note that the entire chain upstream of the lower bound of F_o is suspect. Therefore, the results obtained by [13] are not very useful at improving the average hit index. The key takeaway from this section is that under multiple fault assumption there is no way to determine an upper bound of failures in regular scan chains that is better than the cell closest to the scan-in pin. Therefore, in online diagnosis, we focus only on the lower bounds that can be determined through scan tests.

III. Proposed Scheme of Online Scan Diagnosis

A. Step I: Application of Flush Tests

In the first step, standard flush tests [16] are applied to identify the faulty chains and the fault type.

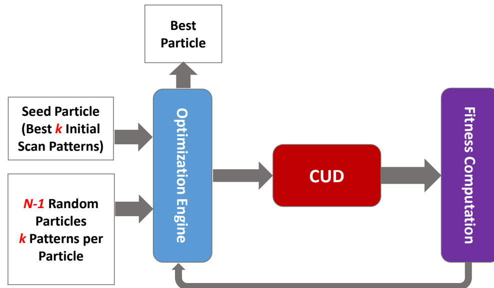

Fig. 6. Overall Scheme of Online Chain Diagnosis

B. Step II: Online Generation of Scan Tests and Diagnosis

Fig. 6 provides a high level view of the second step. We start the process with a set consisting of the *fittest* (the notion of fitness is defined in the next paragraph) k failing patterns from the detection test set and $N-1$ additional sets (or *particles*) of k random patterns each. We apply these particles to the chip under diagnosis (CUD). The optimization engine uses the fitness as a feedback to modify the existing patterns to evolve better diagnostic patterns. After a predefined number of iterations, we have a final best particle that can diagnose the CUD with a higher quality.

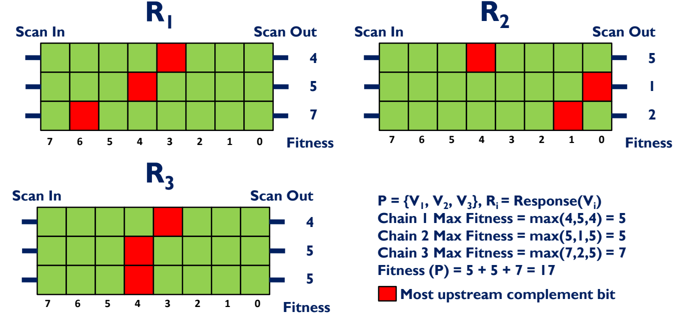

Fig. 7. Fitness Computation

C. Fitness

1) The Most Upstream Complement Bit: We define the most upstream complement bit $b_{muc_{j,i}}$ of a scan chain j as the most upstream bit in j, in a scanned out response R_i whose value is the complement of the scan chain j's stuck-at value as determined by the flush test.

2) Scan Chain Fitness: In the response R_i to the ith scan pattern V_i, we define the distance of the most upstream complement bit $b_{muc_{j,i}}$ from the scan out pin of chain j as the scan chain fitness $f_{sc_{j,i}}$ of chain j under R_i. Under conventional scan cell numbering, where the index 0 is assigned to the cell closest to the scan out, the scan chain fitness for chain j,

$$f_{sc_{j,i}} = index(b_{muc_{j,i}}) + 1$$

The scan chain fitness $f_{sc_{j,i}} = 0$ if there is no complement bit in that chain.

3) Particle Fitness: The fitness f_P of the particle P is the sum of the best fitnesses of the scan chains under the application of its constituent patterns. Mathematically,

$$f_P = \sum_{\forall j \in C} max(f_{sc_{j,i}}, \forall V_i \in P)$$

where, C is the set of scan chains in the circuit. Figure 7 provides an example of the computation of fitness for a particle consisting of three patterns.

4) Lower Bounds Determination: Once we have the final best particle at the end of the optimization run, we can compute the lower bound LB_j for each faulty scan chain j by the following equation.

$$LB_j = b_{muc_j} + 1$$

D. Segmented Approach

The main idea of this design modification is to have the ability to scan out a partial scan chain. For this purpose, we propose to divide the scan chain into p segments and link the scan cell output at the end of each segment to the scan out pin through one or more multiplexers. These multiplexers can be easily instantiated into the design and can be controlled from primary inputs or some configuration register with a much

smaller area overhead than [5]–[8]. Besides, the primary inputs may be shared amongst the multiplexers if there is a constraint on the pin count.

E. Key Advantages

1) Adaptive to Actual Failures: The new patterns generated during the swarm optimization are specific to the diagnosis of the CUD and will be different for the CUDs that fail differently.

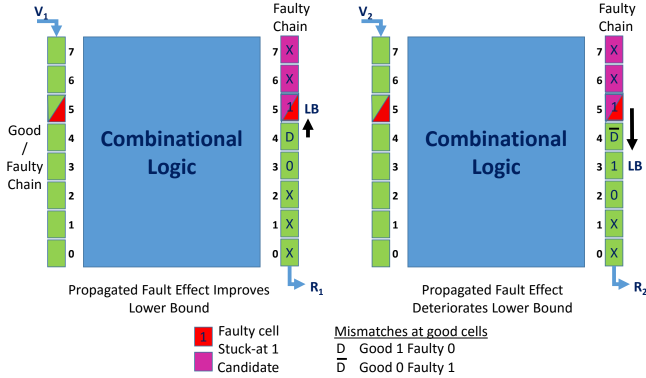

Fig. 8. Possible Effects of Fault Propagation

2) Fault Tolerance: Any logic or input fault can propagate to either a) downstream of b_{muc}, or b) upstream of b_{muc}, or c) to the b_{muc} itself. In the first case, it will not affect diagnosis. In the second case, it will improve diagnosis if its value is the same as b_{muc} (otherwise no effect). In the third and relatively less likely case, it will deteriorate diagnosis (as the faulty value is complement of b_{muc}). The existence of other different patterns in the particle, however, helps mitigate such occurrences. Figure 8 illustrates these effects.

IV. Detail Design

We use binary particle swarm optimization (PSO) to generate new and better test patterns for online scan chain diagnosis. Genetic Algorithms (GA) and Simulated Annealing (SA) are other valid meta-heuristics to optimize functions in large discrete search spaces. We, however, borrow the basic components of our PSO engine from [17] and modify them to suit our requirement. Our particle *swarm* constitutes of N particles, where each particle is a set of k scan patterns. We choose k to be equal to the number of scan chains in the circuit so that we can optimize each scan chain independently.

A. The Particle

A particle is a software object having position, velocity and local best position as its data members.

1) Position: The particle position is a concatenation of k scan patterns, where k is equal to the number of scan chains. Each positional bit can be deemed as a binary position coordinate in an mk dimensional space where m is the number of primary inputs and the scan cells of the circuit under diagnosis (CUD). Mathematically, $\mathbf{P} = [p_0, p_1, ..., p_i, ..., p_{mk-1}]^T$ where $p_i \in \{0, 1\}$.

2) Velocity: The particle velocity is a vector of real numbers having the same size as the position vector. Each component of the vector represents the magnitude of the velocity in the corresponding dimension. Mathematically, $\mathbf{V} = [v_0, v_1, ..., v_i, ..., v_{mk-1}]^T$ such that $v_i \in \mathbb{R}$ and $MINVEL \leq v_i \leq MAXVEL$. In each direction, the velocity is a real number whereas the corresponding position is a bit, the relationship between whom is defined such that the magnitude of velocity is directly proportional to the probability of flipping the positional bit.

3) Local Best Position: We define the local best position of a particle, at any moment during the optimization process, as the position that had the highest fitness in the history of that particle.

B. Global Best Position

We define the global best position, at any moment during the optimization process, as the position of *any* particle that had the highest fitness in the swarm's history until that time.

C. Swarm Initialization

We initialize the swarm with $N - 1$ particles having randomly generated position vectors. The magnitudes of the initial velocity components are chosen from a uniform distribution in the interval $[HIGHVEL,\ MAXVEL]$, whereas their signs are randomly assigned with equal probability. To ensure rapid convergence of the swarm the velocities are initialized in a high range i.e. $HIGHVEL$ is chosen close to $MAXVEL$.

D. Swarm Seeding

Besides the initial random particles, we also seed the swarm with a seed particle, using the k best scan patterns (according to fitness) from the initial detection test set, making the total particle count equal to N.

E. Swarm Iteration

In every iteration, we update the velocities, positions and local best positions of all particles in the swarm. We also update the global best position after all particles have been updated.

1) Velocity Update: The velocity update in the ith dimension is governed by the following equation.

$$v_i[t+1] = c_{iner} v_i[t] + c_{cog}(p_i^{LB}[t] - p_i[t]) \\ + c_{soc}(p^{GB}[t] - p_i[t])$$

Here, $v_i[t+1]$ and $v_i[t]$ are the velocities in the next and present iterations respectively. The constants c_{iner}, c_{cog} and c_{soc} are the inertial, cognitive and social weights. And, $p_i[t]$, $p_i^{LB}[t]$ and $p^{GB}[t]$ are the particle's present position, the particle's local best position and the global best position respectively.

In the above equation we see that the velocity update has three components. 1) The particle's present velocity. 2) The difference between the particle's present position and its local best position. 3) The difference between the particle's present position and the global best position. The relative influence of

these components on the updated velocity is determined by the inertial, cognitive and social weights respectively. In case when the velocity in the next iteration is out of bounds, we limit it to the maximum magnitude (i.e. $v_i[t+1] := MINVEL$ if $v_i[t+1] < MINVEL$ and $v_i[t+1] := MAXVEL$ if $v_i[t+1] > MAXVEL$).

2) Position Update: To update position, we generate a uniformly distributed pseudo-random number x over the interval 0 and $|MAXVEL|$. If our velocity component's magnitude, $|v_i| > x$, we flip the bit of the corresponding position component, i.e., $p_i[t+1] = \sim p_i[t]$. Otherwise, there is no change i.e., $p_i[t+1] = p_i[t]$. This ensures that a higher magnitude of the velocity increases the probability of the position bit being flipped.

3) Fitness Computation and Local and Global Best Update: Once we have determined the position vectors of all particles for the next iteration, we compute the fitness of every particle and update the local and global best position, if needed.

F. The Optimization Algorithm

The overall process for online scan chain diagnosis is formally summarized in Algorithm 1.

Algorithm 1 Proposed Online Diagnosis Algorithm

1: Apply original test set to circuit
2: Use the k best failing patterns to create a seed particle
3: Compute and save the fitness of the seed particle
4: Initialize swarm with the seed particle and $N-1$ random particles
5: **for** $i = 1 : MAX_ITERATIONS$ **do**
6: **for** All particles in swarm **do**
7: Update velocity and position
8: Apply position as scan patterns to the CUD
9: Compute fitness
10: **if** fitness $>$ local best fitness **then**
11: Update local best position and fitness
12: **end if**
13: **if** fitness $>$ global best fitness **then**
14: Update global best position and fitness
15: **end if**
16: **end for**
17: **end for**
18: Compute scan chain lower bounds

V. Evaluation Methodology

A. Generation of Faulty Circuits and Baseline Results

We evaluate our proposed method of online scan chain diagnosis on 5 larger full-scan ISCAS'89 circuits each having 5 scan chains. We create detection test sets for each benchmark using a state of the art industrial tool and generate two populations of faulty circuits with 0-3 and 0-7 faults per scan chain each having 300-600 faulty circuit instances. Then, we perform baseline diagnosis by the same tool using the detection as well as the additional diagnostic patterns in our experiments.

B. Online Diagnosis Configuration

For each faulty circuit instance, we create the seed particle using 5 fittest failing patterns from the detection test set. Then, we initialize a swarm of 3 particles with random positions and velocities such that we have 4 total particles including the seed. Setting $MINVEL = -50000$, $MAXVEL = 50000$ and $HIVEL = 48000$ produces the initial velocities within the top 4% of the velocity range. We then run the online diagnosis with $MAX_ITERATIONS = 5$ to evolve and save the best particle. During this process, we set $c_{cog} = 20000$, $c_{soc} = 20000$ and linearly vary c_{iner} from 8000 to 100 over 1 to $MAX_ITERATIONS$. All computations are done using suitably scaled integers and no floating point operations are involved.

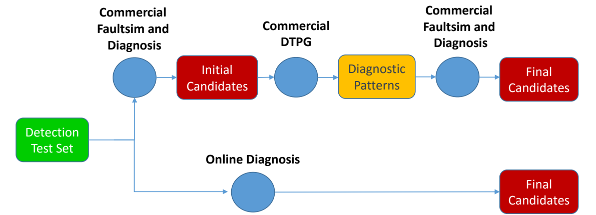

Fig. 9. A simplified schematic of the evaluation process

C. Comparative Analysis

The upper and lower bounds of the suspect range in the baseline results are non-definitive, which means the actual faults may be found beyond both ends of the suspect range. We believe that this non-definitiveness is because the industrial tool most likely uses a dictionary based approach such as [14]. On the other hand, the list of suspects from the online diagnosis contains all the cells upstream from the chain lower bound.

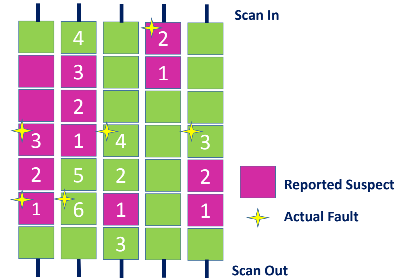

Fig. 10. Counting Hit Index Beyond the Suspect List

To have a fair comparison, we continue to count the hit index beyond the suspect list reported by the offline tool. Figure 10 illustrates the counting method. To start with, we count all the suspects in the reported list from the top ranking to the lowest ranking one. Then, we count alternately on both sides of the suspect list until we hit all inserted faults in the suspect list. The pink cells are the reported suspects and the stars are the actual inserted faults, whereas the numbers are the counted indices. E.g. in the second chain from the left, we first count the reported suspects up to '3' within the suspect range and then on alternate sides i.e. '4' upstream and '5' downstream. As there is no more cells upstream, we continue

to count downstream and hit the fault at '6', which becomes the inserted fault's hit index. An alternative to this counting method is to continue counting upstream all the way until we reach the end of the scan chain and then continue downstream. This, however, yielded worse results for the offline diagnosis.

VI. Results

A. Accuracy Comparison

The accuracy of the online method is always 100% because the lower bound is definitive. The accuracy of offline diagnosis, on the other hand, is usually less than 100% because many faults miss the reported suspect list. However, due to the adjustment made for a fair comparison in the previous section, we effectively include the entire faulty chain in the list of offline diagnosis suspects. Resultantly, the accuracy is marked at 100% for both techniques and not reported here.

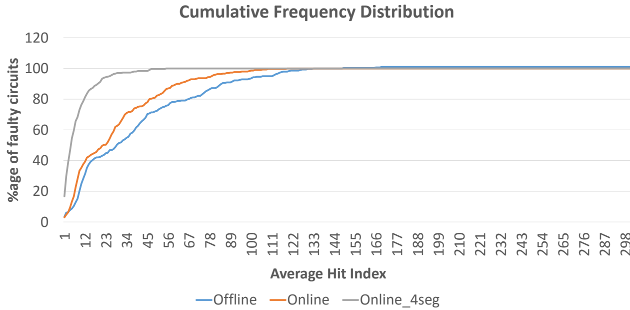

Fig. 11. Comparison of Average Hit Index for Circuits having Max 3 Faults Per Chain

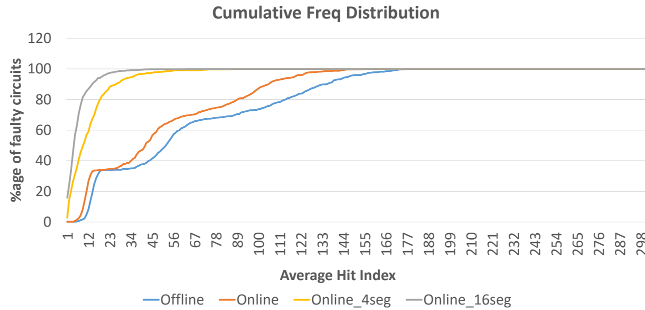

Fig. 12. Comparison of Average Hit Index for Circuits having Max 7 Faults Per Chain

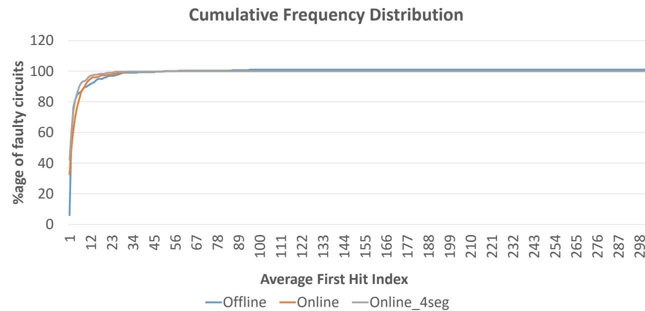

Fig. 13. Comparison of Average First Hit Index for Circuits having Max 3 Faults Per Chain

B. Hit Index Comparison

The graphs in Figure 11 and 12 plot the cumulative frequency distribution of the average hit index for the offline

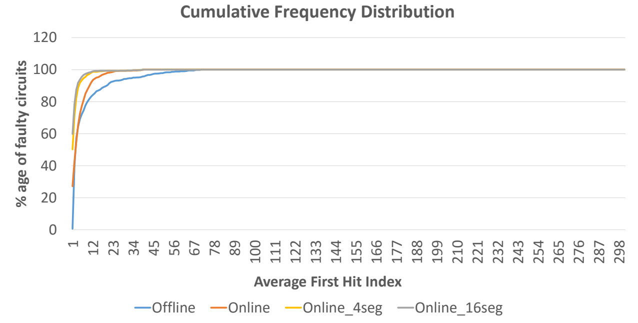

Fig. 14. Comparison of Average First Hit Index for Circuits having Max 7 Faults Per Chain

and two different configurations of the online diagnosis for populations of faulty chips having at most 3 and 7 faults per chain respectively. Again, a point (x, y) on these graphs means that $y\%$ faulty chips were diagnosed with an average hit index of less than or equal to x. Figure 13 and 14 show similar plots for the average first hit indices for populations of faulty chips having at most 3 and 7 faults per chain respectively. It may be observed that the online diagnosis outperforms the offline diagnosis in all populations for both the segmented and unsegmented cases.

C. Benchmark-wise Results

The table I presents the benchmark wise aggregated results for average hit index and average first hit index for different techniques and configurations. The first and second columns state the faults inserted per chain (FPC) in the population and the names of benchmarks. Columns 3 and 4 list the population averages of the average hit index (HI) and average first hit index (FHI) achieved by the offline diagnosis using the initial test set. Columns 5 and 6 present the HI and FHI for the case when diagnostic patterns were generated by the industrial tool. Columns 7-10 and 11-14 present the same results for online diagnosis with a swarm having 4 particles, 5 patterns per particle and 5 iterations and 8 particles, 5 patterns per particle and 5 iterations respectively. For the 0-3 FPC case, we present results using 4 segments, whereas for the 0-7 FPC case the results are presented for 16 segments. The online method outperforms the state-of-the-art offline method for all benchmarks, in terms of the average hit index for all faults inserted into circuits, even when the chains were not segmented. The results for the segmented approach are even better. In terms of the average first hit index, we do see the offline method performing better than the online for a couple of benchmarks. For the remaining benchmarks, the online method is better, even when we are comparing the first hit indices.

D. Computational Complexity

The computational complexity of our method is $O(k \times p \times N \times MAX_ITERATIONS \times G/10) = O(G)$, where G is the number of gates in the circuit and we assume that the number of scan elements and primary inputs of the circuit is about $G/10$. The other quantities have been defined earlier in the paper. We have observed in our experiments that, for the scan chain diagnosis, small sized swarms can converge to an acceptable quality in significantly fewer iterations than the logic diagnostic test pattern generator (DTPG) proposed in

TABLE I. CIRCUIT-WISE COMPARISON OF AVERAGE HIT INDEX AND AVERAGE FIRST HIT INDEX

FPC	Bench	Offline				Online (4, 5, 5)				Online (8, 5, 5)			
		Initial Test Set		with Addl Patterns		Full Chain		Segmented Chain		Full Chain		Segmented Chain	
		Avg HI	Avg FHI	Avg HI	Avg FHI	Avg HI	Avg FHI	Avg HI	Avg FHI	Avg HI	Avg FHI	Avg HI	Avg FHI
0-3	s05378	10.08	1.89	10.07	1.88	6.78	1.08	1.46	1.1	6.78	1.08	1.42	1.06
	s09234	8.14	2.15	8.13	2.14	7.12	2.66	3.52	2.3	7.1	2.64	3.46	2.32
	s13207	36.01	4.67	36.01	4.67	27.42	5.5	10.04	3.86	27.44	5.52	9.94	3.86
	s15850	31.19	5.47	31.90	5.99	22.9	4.44	6.72	3.48	22.8	4.32	6.68	3.48
	s38417	81.99	11.49	81.64	10.97	58.78	8.6	17.4	6.14	58.66	8.56	17.56	6.1
	s38584	65.22	4.37	65.21	4.34	43.44	3.5	8.28	2.74	43.46	3.54	8.36	2.68
0-7	s05378	14.31	2.28	14.32	2.28	10.75	1.1	1.18	1.04	10.74	1.1	1.18	1.04
	s09234	12.76	2.17	12.78	2.23	11.02	2.33	3.74	1.63	11.05	2.34	3.82	1.66
	s13207	54.20	8.27	54.17	8.15	44.84	6.13	9.14	2.64	44.73	6.04	9.16	2.6
	s15850	48.21	6.82	48.19	6.76	38.54	4.79	6.04	2.03	38.49	4.68	5.99	2.03
	s38417	132.35	18.45	132.25	18.28	103.98	9.96	13.73	4.2	103.95	9.93	13.56	4
	s38584	108.38	7.05	108.25	6.85	84.81	4.36	6.23	1.69	84.76	4.35	6.19	2.01

[17]. This is because there are many available ways of placing the complement of a stuck-at value in a scan chain or segment. The computational complexity for the offline approach, on the other hand, is $O(G^2)$ if it is fault simulation based and even higher if it is deterministic.

VII. CONCLUSION

We have proposed an effective online scan chain diagnosis method for multiple stuck-at failures on scan chains that works by iteratively modifying test patterns, and applying them to the chip under diagnosis. The method is adaptive to the actual defect(s) and eliminates the need for offline analysis. In our experiments on the larger ISCAS'89 benchmarks, we achieved 70% and 37% improvement in the average avg. hit index over all circuit instances, for the segmented and non-segmented cases respectively, as compared to a state-of-the-art offline industry tool, when 0 to 7 faults were randomly inserted in each scan chain. Our method does require additional tester time, which may be preferred to the computational, setup and overhead costs of the offline diagnosis during the yield learning process.

REFERENCES

[1] S. Kundu, "On diagnosis of faults in a scan-chain," in *VLSI Test Symposium, 1993. Digest of Papers., Eleventh Annual 1993 IEEE*. IEEE, 1993, pp. 303–308.

[2] Y. Huang, R. Guo, W.-T. Cheng, and J. C.-M. Li, "Survey of scan chain diagnosis," *IEEE Design & Test of Computers*, vol. 25, no. 3, pp. 240–248, 2008.

[3] K. De and A. Gunda, "Failure analysis for full-scan circuits," in *Test Conference, 1995. Proceedings., International*. IEEE, 1995, pp. 636–645.

[4] P. Song, F. Stellari, T. Xia, and A. J. Weger, "A novel scan chain diagnostics technique based on light emission from leakage current," in *Test Conference, 2004. Proceedings. ITC 2004. International*. IEEE, 2004, pp. 140–147.

[5] J. L. Schafer, F. A. Policastri, and R. J. McNulty, "Partner srls for improved shift register diagnostics," in *VLSI Test Symposium, 1992.'10th Anniversary. Design, Test and Application: ASICs and Systems-on-a-Chip', Digest of Papers., 1992 IEEE*. IEEE, 1992, pp. 198–201.

[6] S. Edirisooriya and G. Edirisooriya, "Diagnosis of scan path failures," in *Proceedings of the 13th IEEE VLSI Test Symposium*. IEEE Computer Society, 1995, p. 250.

[7] S. Narayanan and A. Das, "An efficient scheme to diagnose scan chains," in *Test Conference, 1997. Proceedings., International*. IEEE, 1997, pp. 704–713.

[8] ——, "Flip-flop design and technique for scan chain diagnosis," Mar. 9 1999, uS Patent 5,881,067.

[9] R. Guo and S. Venkataraman, "A new technique for scan chain failure diagnosis," in *ISTFA 2002: 28 th International Symposium for Testing and Failure Analysis*, 2002, pp. 723–732.

[10] ——, "An algorithmic technique for diagnosis of faulty scan chains," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 9, pp. 1861–1868, 2006.

[11] Y.-L. Kao, W.-S. Chuang, and J. C.-M. Li, "Jump simulation: a technique for fast and precise scan chain fault diagnosis," in *2006 IEEE International Test Conference*. IEEE, 2006, pp. 1–9.

[12] Y. Huang, "Dynamic learning based scan chain diagnosis," in *Proceedings of the conference on Design, automation and test in Europe*. EDA Consortium, 2007, pp. 510–515.

[13] Y. Huang, W.-T. Cheng, and R. Guo, "Diagnose multiple stuck-at scan chain faults," in *Test Symposium, 2008 13th European*. IEEE, 2008, pp. 105–110.

[14] R. Guo, Y. Huang, and W.-T. Cheng, "Fault dictionary based scan chain failure diagnosis," in *16th Asian Test Symposium (ATS 2007)*. IEEE, 2007, pp. 45–52.

[15] L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI test principles and architectures: design for testability*. Elsevier Inc., 2006.

[16] Y. Wu, "Diagnosis of scan chain failures," in *Defect and Fault Tolerance in VLSI Systems, 1998. Proceedings., 1998 IEEE International Symposium on*. IEEE, 1998, pp. 217–222.

[17] S. Tanwir, M. S. Hsiao, and L. Lingappan, "Hardware-in-the-loop model-less diagnostic test generation," in *High Level Design Validation and Test Workshop (HLDVT), 2016 IEEE International*. IEEE, 2016, pp. 128–135.